

Energy Consumption Forecasting using Ensemble Learning Algorithms

Jose Silva¹, Isabel Praça¹, Tiago Pinto¹ and Zita Vale²

¹GECAD research group, Institute of Engineering, Polytechnic of Porto (ISEP/IPP),
Porto, Portugal

{icp, tcp}@isep.ipp.pt

² Polytechnic of Porto (ISEP/IPP), Porto, Portugal
zav@isep.ipp.pt

Abstract. The increase of renewable energy sources of intermittent nature has brought several new challenges for power and energy systems. In order to deal with the variability from the generation side, there is the need to balance it by managing consumption appropriately. Forecasting energy consumption becomes, therefore, more relevant than ever. This paper presents and compares three different ensemble learning methods, namely random forests, gradient boosted regression trees and Adaboost. Hour-ahead electricity load forecasts are presented for the building N of GECAD at ISEP campus. The performance of the forecasting models is assessed, and results show that the Adaboost model is superior to the other considered models for the one-hour ahead forecasts. The results of this study compared to previous works indicates that ensemble learning methods are a viable choice for short-term load forecast.

Keywords: Electricity consumption, Short-term load forecast, Ensemble learning methods, Forecasting

1 Introduction

Electricity demand forecasting is an important task for the agents and companies involved in the electricity market. The features of the electricity, that is a non-storable product, and also the rules of this competitive market create the need of accurate predictions of electricity demand in order to anticipate decisions [1]. Thus, Electric Power Load Forecasting (EPLF) is a crucial process in the planning of electricity industry and the operation of electric power systems. The EPLF is classified in terms of the planning horizon's duration up: 1 day for short-term load forecasting (STLF), 1 day to 1 year for medium-term load forecasting (MTLF), and 1-10 years for long-term load forecasting (LTFL) [2]. A vast number of studies have developed accurate models in recent years. There are usually two approaches. The traditional statistical approach like multiple linear regression (MLR) used in [3] that achieved a 3.99% Mean Absolute Percentage Error (MAPE) on hourly electric load forecast. However, traditional statistical methods often result in lower accuracy because they are inadequate to fully model the complex nature of electricity demand. Artificial Intelligence (AI) techniques are more reliable due to their ability to identify non-linear relationships between dependent and independent variables. Artificial Neural Networks (ANN) [4], Support

Vector Machines (SVM) [5], Random Forest [6] and Stochastic Gradient Boosting [7] are popular AI techniques for STLF.

This paper presents and compares the application of three ensemble learning methodologies to the problem of STLF. The applied algorithms are Random Forests (RF), Gradient Boosted Regression Trees (GBR) and Adaboost (AR2). These algorithms are adapted and applied to the forecasting of energy consumption of an office building. The objective of this study is to forecast a better profile of the energy consumption for the coming hours. Results from the proposed approach are compared to those achieved in previous studies, using different techniques, namely three fuzzy based systems: an Hybrid Neural Fuzzy Inference System (HyFIS) [8], the Wang and Mendel's Fuzzy Rule Learning Method (WM) [9]; and SVM [10]. The case study is based on real data referring to the electricity consumption of a campus building of ISEP/GECAD - Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development.

After this introductory section, section II presents the formulation and explanation of the proposed approach, section III presents the achieved results and discusses their comparison to the results achieved by previous methods. Section IV presents the most relevant conclusions and contributions of this work

2 Material and methods

This paper presents and discusses the implementation of three ensemble-learning methods to forecast the electricity consumption of an office building. The electricity consumption from building N of the GECAD research center located in ISEP/IPP, Porto, Portugal is used in this work. The ensemble approach has been developed based on Python programming language. The implementation details and results of this work are discussed and compared in the following sections.

Ensemble methods combine the predictions of several base estimators built with a given learning algorithm in order to improve generality and robustness over a single estimator. They can be distinguished in two categories; averaging methods, where the main idea is to build several estimators independently and then to average their predictions, and boosting methods where the driving principle is to combine several weak models to produce a powerful ensemble. Examples of the first category are the bagging methods and RF while the second category includes methods like Adaboost and GBRT.

2.1 Random Forests (RF)

Random forests is an ensemble learning method for classification and regression. In RF each tree in ensemble is generated by randomly selecting the attributes to split at each node and these features on training set are used to estimate best split. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, often more than compensating for the increase in bias, resulting an overall better model. Let us assume that, the training data contain K instances and M set of features. We are

given n , where n is the set of variable $n \in N$. Learning set is created by choosing k instances from K with replacement, and the remaining instances are used to estimate the error of the model. At each node, randomly select n variables and make split decisions based on these variables. At the end fully grown unpruned tree is created. In this study we used the implementation of Python's scikit-learn library [11], which combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class. For our forecasting task we built a RF regressor of 200 trees with nodes which were expanded until all leaves were pure. The minimum number of samples required to split an internal node has been set to two.

2.2 Gradient Boosted Regression Trees (GBR)

GBRT is a machine learning technique that generates a prediction model in the form of an ensemble of weak prediction models which are typically decision trees. It builds the model sequentially and generalizes them by allowing optimization of an arbitrary differentiable loss function. It is mainly a regression technique invented by Jerome H. Friedman in 1999 [12]. GBR takes into account additive models of the form (1):

$$F(x) = \sum_{m=1}^M \gamma m h_m(x) \quad (1)$$

where $h_m(x)$ are the principle functions, which are called weak learners in the context of boosting and γm the step length that is chosen using line search (2):

$$\gamma m = \underset{\gamma}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) - \gamma \frac{\theta L(y_i, F_{m-1}(x_i))}{\theta F_{m-1}(x_i)}) \quad (2)$$

Similarly to other boosting algorithms GBRT builds the additive model in a forward stage wise approach (3):

$$F_m(x) = F_{m-1}(x) + \gamma m h_m(x) \quad (3)$$

At each stage the decision tree $h_m(x)$ is chosen to minimize the loss function L given the current model F_{m-1} and its fit $F_{m-1}(x_i)$, as in (4).

$$F_m(x) = F_{m-1}(x) + \underset{h}{\operatorname{argmin}} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) - h(x)) \quad (4)$$

The basic idea for solving this minimization problem is to use the steepest descent which is the negative gradient of the loss function evaluated at the current model F_{m-1} . This can be estimated as in (5):

$$F_m(x) = F_{m-1}(x) + \gamma m \sum_{i=1}^n \nabla_F L(y_i, F_{m-1}(x_i)) \quad (5)$$

To forecast our target values, we built a GBRT model with 1400 boosting stages using Python's scikit-learn package. Since gradient boosting is fairly robust to over-

fitting, a large number of stages usually performs better. Scikit-learn supports many different loss function. We experienced all of them and found that the Least absolute deviations (LAD) function works better for our model.

To improve the performance of our model we also tuned the maximum depth of the individual regression estimators. This depth limits the number of nodes in the tree. Maximum depth of size 10 produced the best results. In addition, we set the minimum number of samples required to split an internal node to 2 and the learning rate to 0.2. The learning rate scales the step length of the gradient decent procedure. The parameter v in the following equation is the learning rate:

$$Fm(x) = Fm - 1(x) + v\gamma mhm(x) \quad (6)$$

The above regularization technique was proposed in [13] and scales the contribution of each weak learner by v .

2.3 AdaBoost

Another used ensemble estimator is AdaBoost.R2 [14] which is a modified regression version of the famous AdaBoost ensemble estimator [15]. It sequentially fits estimators and each subsequent estimator concentrates on the samples that were predicted with higher loss.

The used algorithm implemented in [11] slightly differs from [14] as it allows to use the weights directly in the fitted estimator and not only for weighted sampling of features, as follows:

1. start algorithm $t = 0$
2. To each training sample assign initial weight (7)

$$w_i^t := 1, i = 1, 2, \dots, m \quad (7)$$

3. fit estimator t to the weighted training set with weights w_i^t
4. compute prediction \hat{y}_i^t using the estimator t for each sample i
5. compute loss l_i for each training sample (8)

$$l_i^t = \text{loss}(|\hat{y}_i^t - y_i|) \quad (8)$$

6. calculate average loss \bar{l}^t
7. calculate confidence β^t for the estimator (low β^t means high confidence in estimator t)
8. update weights of training samples (9)

$$w_i^{t+1} = w_i^t \cdot (\beta^t)^{(1-l_i^t)}, i = 1, 2, \dots, m \quad (9)$$

9. $t = t + 1$ continue to step 3 while the average loss $\bar{l}^t < 0.5$

To forecast our target values, we built an AR2 model with 1400 boosting stages using Python's scikit-learn package and the learning rate has been set to 0.01. We experienced that square loss function works better for our model [14] (10):

$$l_i^t = \frac{|\hat{y}_i^t - y_i|^2}{D^2} \quad (10)$$

where D is defined as (11):

$$D = \sup\{|\hat{y}_i^t - y_i|, i \in \{1, 2, \dots, m\}\} \quad (11)$$

3 Case study

3.1 Dataset description

This study uses real data from ISEP/GECAD building N monitoring, an office building with a daily usage from around 30 researchers. The data is available in a SQL server that stores the information given by five energy analyzers, each of them stores the electricity consumption data coming from sockets, lighting and HAVAC from a part of the building, in a 10 seconds time interval. A java base application has been developed in this implementation. Which collect the data from the SQL server and calculate the average of the total electricity consumption of the building N – ISEP/GECAD per each hour. This application also creates a new .csv file in a format that can be used as the input of the forecast method.

The ensemble methods have been implemented for this forecast with various strategies presented in Table 2. The various strategies combine the features extracted from the 10 days before the hour which is meant to be forecasted presented in A total of 15 features has been generated. Table 1 presents the features used for training and testing of the building N consumption dataset.

Table 1. The results of this method are shown and compared to those of previous works.

3.2 Results

In order to test these methods, the electricity consumption from 00:00 until 23:00 of the date 5/4/2018 is forecasted. MAPE is used for error calculation as means to compare the forecasted values and the real values of each hour.

A total of 15 features has been generated. Table 1 presents the features used for training and testing of the building N consumption dataset.

Table 1- Generated Features

Feature Description	Nomenclature
Consumption of the 3 previous hours	Z_{t-1} to Z_{t-3}
Hour of the day	H_t
Month of the year	M_t
Day of the Month	D_t
Year	Y_t
Day of the Week	DoW_t
Environmental temperature of the hour (°C)	T_t
Environmental temperature from the previous 3 hours	T_{t-1} to T_{t-3}
Consumption at the same hour from the 2 previous weeks	Z_{t-168} and Z_{t-336}
Environmental Humidity of the hour	Hu_t

The combination of the generated features in Table 1 resulted in the creation of seven training strategies, which are presented in Table 2.

Strategy #	$Z \sim * (\text{Consumption over } \dots)$
1	$Z \sim D_t * M_t * H_t * Y_t * DoW_t$
2	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t$
3	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t * Z_{t-1} \text{ to } Z_{t-3}$
4	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t * Z_{t-1} \text{ to } Z_{t-3} * T_{t-1} \text{ to } T_{t-3}$
5	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t * Z_{t-168} \text{ and } Z_{t-336}$
6	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t * Z_{t-1} \text{ to } Z_{t-3} * T_{t-1} \text{ to } T_{t-3} * Z_{t-168} \text{ and } Z_{t-336}$
7	$Z \sim D_t * M_t * H_t * Y_t * DoW_t * T_t * Hu_t$

In Table 2 the strategies were designed to test the performance of different combinations of variables. The MAPE error results for each ensemble method and each strategy are presented in Figure 1.

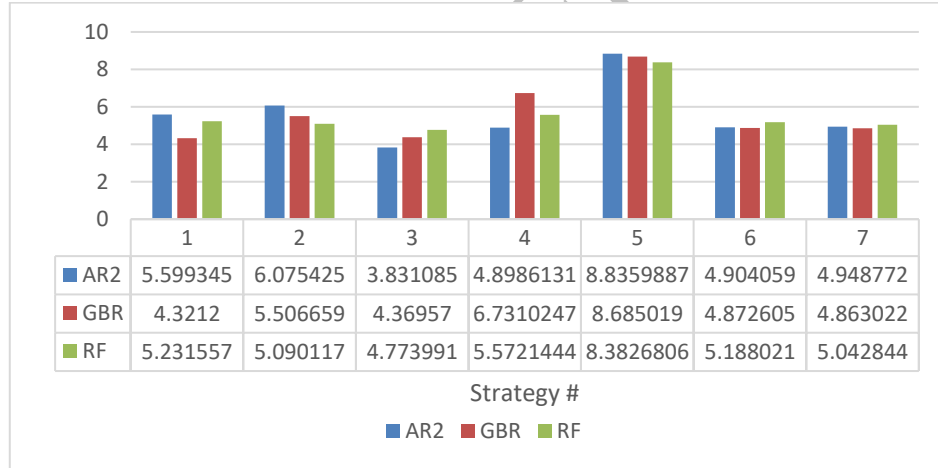


Figure 1 - Average forecasting errors of the AR2, GBR and RF

As Figure 1 shows, the forecasting error of the third training strategy presents the best performance in all of the ensemble methods. It is noticeable that the use of the environmental temperature is better in all the scenarios but the lagged features of the environmental temperature produces a higher error than not using it. In addition, the use of lagged features of consumption performs better than not using it, the same happens to the environmental humidity.

The study presented in [16] addresses the electricity consumption forecast based on fuzzy rules methods of the same location as this study, namely using HyFIS, WM and SVM. In order to compare the results of the ensemble methods with the ones addressed in [16] we trained our models using the third training strategy and forecasted the 24

hours of 10/4/2018. The comparison between the methods is shown in Figure 2 and Table 3.

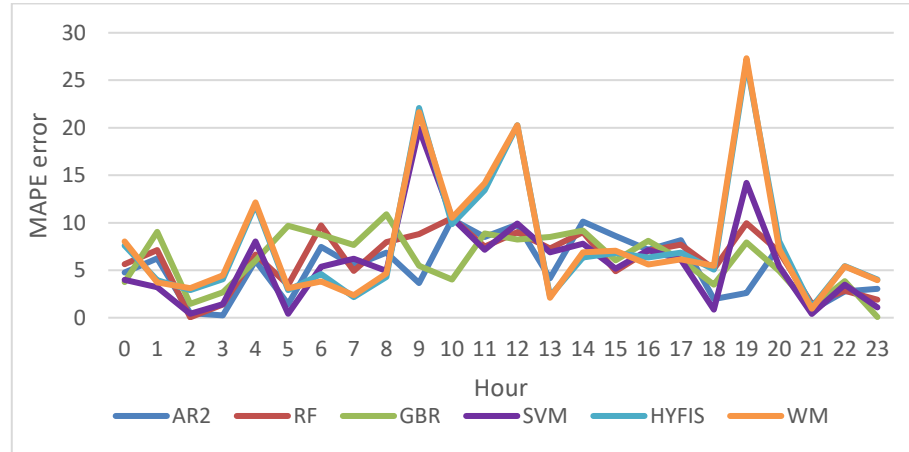


Figure 2 - Comparison between the results of AR2, RF and GBR and the results in [16]

Table 3 - Average forecasting methods errors

	AR2	RF	GBR	SVM	HyFIS	WM
MAPE %	5,34	6,11	6,07	5,82	7,88	7,92

The comparison between the fuzzy rule based methods and the ensemble methods in Figure 2 and Table 3 shows that the AR2 method has the lower average forecasting error between these methods and can achieve better forecasting results than the other methods.

4 Conclusions

This paper addresses the use of ensemble learning methodologies to forecast the electricity consumption of an office building in the following hours. This method uses the value of the electricity consumption from previous days to preview this value for the next hour.

By comparing the results of the AR2 method to the results of the SVM and some fuzzy rule-based methods, it is possible to conclude that it provides a more accurate consumption forecast. Additionally, the results presented in this paper in comparison to the results of the fuzzy rule-based methods presented in [16], namely SVM, HyFIS and WM, tend to forecast more reliable values and all the calculated errors are closer to the average error with exception to SVM that outperforms RF and GBR. The use of environmental variables such as a humidity and temperature proved to be useful in this experiment as well as the use of lagged features of previous hours.

As future work, we suggest the inclusion of additional exogenous variables in our models, such as direct solar irradiation or thermal sensation in order to improve the forecast accuracy.

Acknowledgements

This work has received funding from National Funds through FCT (Fundação da Ciencia e Tecnologia) under the project SPET – 29165, call SAICT 2017

References

1. Zhang X, Wang J, Zhang K (2017) Short-term electric load forecasting based on singular spectrum analysis and support vector machine optimized by Cuckoo search algorithm. *Electr Power Syst Res* 146:270–285. <https://doi.org/10.1016/j.epsr.2017.01.035>
2. Raza MQ, Khosravi A (2015) A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew Sustain Energy Rev* 50:1352–1372. <https://doi.org/10.1016/j.rser.2015.04.065>
3. Saber AY, Alam AKMR (2017) Short term load forecasting using multiple linear regression for big data. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp 1–6
4. Pinto T, Sousa TM, Vale Z (2012) Dynamic artificial neural network for electricity market prices forecast. *Intell. Eng. Syst. (INES)*, 2012 IEEE 16th Int. Conf. 311–316
5. Pinto T, Sousa TM, Praça I, et al (2016) Support Vector Machines for decision support in electricity markets' strategic bidding. *Neurocomputing* 172:438–445. <https://doi.org/10.1016/j.neucom.2015.03.102>
6. Ahmad T, Chen H (2019) Nonlinear autoregressive and random forest approaches to forecasting electricity load for utility energy management systems. *Sustain Cities Soc* 45:460–473. <https://doi.org/10.1016/j.scs.2018.12.013>
7. Touzani S, Granderson J, Fernandes S (2018) Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy Build* 158:1533–1543. <https://doi.org/10.1016/j.enbuild.2017.11.039>
8. Osório GJ, Matias JCO, Catalão JPS (2015) Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information. *Renew Energy* 75:301–307. <https://doi.org/10.1016/j.renene.2014.09.058>
9. Gou J, Hou F, Chen W, et al (2015) Improving Wang–Mendel method performance in fuzzy rules generation using the fuzzy C-means clustering algorithm. *Neurocomputing* 151:1293–1304. <https://doi.org/10.1016/j.neucom.2014.10.077>
10. Du P, Wang J, Yang W, Niu T (2018) Multi-step ahead forecasting in electrical power system using a hybrid forecasting system. *Renew Energy* 122:533–550. <https://doi.org/10.1016/j.renene.2018.01.113>
11. Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 12:2825–2830
12. Friedman JH (2002) Stochastic Gradient Boosting. *Comput Stat Data Anal* 38:367–378. [https://doi.org/10.1016/S0167-9473\(01\)00065-2](https://doi.org/10.1016/S0167-9473(01)00065-2)
13. Friedman JH (2001) Greedy Function Approximation: A Gradient Boosting Machine. *Ann Stat* 29:1189–1232
14. Drucker H (1997) Improving Regressors Using Boosting Techniques. In: *Proceedings of the Fourteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 107–115
15. Freund Y, Schapire RE (1997) A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J Comput Syst Sci* 55:119–139. <https://doi.org/10.1006/jcss.1997.1504>
16. Jozi A, Pinto T, Praça I, Vale Z (2018) Day-ahead forecasting approach for energy consumption of an office building using support vector machines. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp 1620–1625